

The background is a blue gradient with decorative white circuit-like lines in the corners. These lines consist of straight segments and small circles, resembling a stylized electronic circuit or data flow diagram.

DATA MASTERY WITH SQL: CRAFTING EFFICIENT QUERIES

INPUT QUERY

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1 --Retrieve the total number of orders placed
2
3 select count(*) from orders|
```

Data Output Messages Notifications

	count bigint	
1	21350	

OUTPUT

INPUT QUERY

OUTPUT

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1 --Calculate the total revenue generated from pizza sales
2
3 select sum((order_details.quantity * pizzas.price)) as Total_Revenue
4 from
5 order_details
6 join pizzas on order_details.pizza_id=pizzas.pizza_id
```

Data Output Messages Notifications

	total_revenue numeric
1	817860.05

INPUT QUERY

OUTPUT

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1  -- Identify the highest-priced pizza
2
3  select pizza_types.name,pizzas.price
4  from pizza_types join pizzas
5  on pizza_types.pizza_type_id=pizzas.pizza_type_id
6  order by price desc limit 1
```

Data Output Messages Notifications

	name character varying (50)	price numeric (10,2)
1	The Greek Pizza	35.95

INPUT QUERY

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1 -- List the top 5 most ordered pizza types along with their quantities
2 select pizza_types.name,
3 sum(order_details.quantity)
4 from pizza_types join pizzas
5 on pizza_types.pizza_type_id=pizzas.pizza_type_id
6 join order_details
7 on order_details.pizza_id=pizzas.pizza_id
8 group by name
9 order by sum desc limit 5
```

Data Output Messages Notifications

	name character varying (50)	sum bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418

OUTPUT

INPUT QUERY

OUTPUT

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2
3  select pizza_types.category,
4  sum(order_details.quantity) from pizza_types
5  join pizzas
6  on pizza_types.pizza_type_id=pizzas.pizza_type_id
7  join order_details
8  on order_details.pizza_id=pizzas.pizza_id
9  group by category order by sum(order_details.quantity) desc
```

Data Output Messages Notifications

	category character varying (50)	sum bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

INPUT QUERY

OUTPUT

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2
3  select round(avg(quantity),0) as avg_pizza_ordered_per_day
4      from
5      (select orders.order_date,sum(order_details.quantity) as quantity
6      from
7      orders join order_details
8      on orders.order_id=order_details.order_id
9      group by order_date)
```

Data Output Messages Notifications

	avg_pizza_ordered_per_day	
	numeric	
1		138

INPUT QUERY

OUTPUT

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1 -- Determine the top 3 most ordered pizza types based on revenue.
2
3 select pizza_types.name, sum(order_details.quantity*pizzas.price) as Revenue
4 from pizza_types join pizzas
5 on pizza_types.pizza_type_id=pizzas.pizza_type_id
6 join order_details
7 on order_details.pizza_id=pizzas.pizza_id
8 group by name order by revenue desc limit 3
```

Data Output Messages Notifications

	name character varying (50)	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50

INPUT QUERY

OUTPUT

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1 -- Calculate the percentage contribution of each pizza type to total revenue.
2 select pizza_types.category, round(sum(order_details.quantity*pizzas.price)/(select
3     sum(order_details.quantity*pizzas.price) from order_details
4     join pizzas on
5     order_details.pizza_id=pizzas.pizza_id)*100,2) as Revenue_Percentage
6 from pizza_types join pizzas
7 on pizza_types.pizza_type_id=pizzas.pizza_type_id
8 join order_details
9 on order_details.pizza_id=pizzas.pizza_id
10 group by category order by Revenue_Percentage desc limit 4
```

Data Output Messages Notifications

	category character varying (50)	revenue_percentage numeric
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

INPUT QUERY

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1 -- Analyze the cumulative revenue generated over time.
2 select revenue_table.order_date,
3       sum(revenue) over(order by revenue_table.order_date) as Cum_revenue
4 from
5       (select orders.order_date, sum(order_details.quantity*pizzas.price) as revenue
6 from orders join order_details
7 on orders.order_id=order_details.order_id
8 join pizzas
9 on pizzas.pizza_id=order_details.pizza_id
10 group by orders.order_date order by orders.order_date) as revenue_table
```

Data Output Messages Notifications

	order_date date	cum_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.60

OUTPUT

INPUT QUERY

Pizzahut/postgres@PostgreSQL 16

Query Query History

```
1 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 v select category,name,revenue from
3 (select category,name,revenue,
4      rank() over(partition by category order by revenue desc) as rank
5  from
6  (select pizza_types.category,pizza_types.name,
7       sum(order_details.quantity*pizzas.price) as revenue
8  from pizza_types join pizzas
9  on pizza_types.pizza_type_id=pizzas.pizza_type_id
10 join order_details
11 on order_details.pizza_id=pizzas.pizza_id
12 group by pizza_types.category,pizza_types.name) )
13 where rank<=3
```

Data Output Messages Notifications

	category character varying (50)	name character varying (50)	revenue numeric
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	12768.00

Total rows: 12 of 12 Query complete 00:00:00.322 Ln 8, 0

OUTPUT